Design Review:
Process Agnostic Library Porting

Joe Murray, Lijun Li

Introduction

For our project, we chose to produce a process agnostic standard library and a program to port this library to any chosen technology based on design rules. This project holds a lot of promise as a useful tool for research. One of the major hurdles in the ever decreasing scale of VLSI manufacturing technologies is the porting of existing circuits and libraries to these new technologies in order to make use of its advantages. Each manufacturing technology step requires a corresponding change in at least circuit design if not a complete library overhaul. This project proposes to help overcome this constraint by producing a technology independent, representative section of a standard library and using this to demonstrate a migration algorithm/program. However, implicit in this approach is a trade off of some optimization parameters for decreased "time to market". To overcome this we proposed to implement some level of optimization to the final layout. This design review will summarize some of the literature on this topic, discuss the results of our first efforts, give a plan for the continuation of this project and give the targeted goals.

Publication review

During our literature review we found a number of useful papers. We found there to be quite a bit of literature on automated circuit migration. Below is a short summary of some of the literature we surveyed.

Interconnect-driven Cell-based Migration of Integrated Circuit Layout, Evgeny Shaphir, Research Thesis

In this recent (2009) research thesis, Evgeny Shaphir describes their method for porting of a micro processor from 65nm to 45nm technology. It starts by giving a brief overview and introduction to process migration including the motivation and other approaches. It then goes on to describe in detail the algorithm used. The algorithm first identified sub-cells and interconnection nodes. The cells were then individually shrunk and an optimization was preformed. The main problem to solve involved the rewiring of the circuit do to the asymmetric shrinking of the wires. The solution implemented simply increased the layers utilized by the layout cell or relaxing wire size constraints. Layout errors that could not be solved were hand corrected. This paper goes extensively into the details of this process.

Their target delay was 0.7 (scaling factor) of the original delay. Below is a graph of their delay results with significant paths that do not meet their target circled in red.

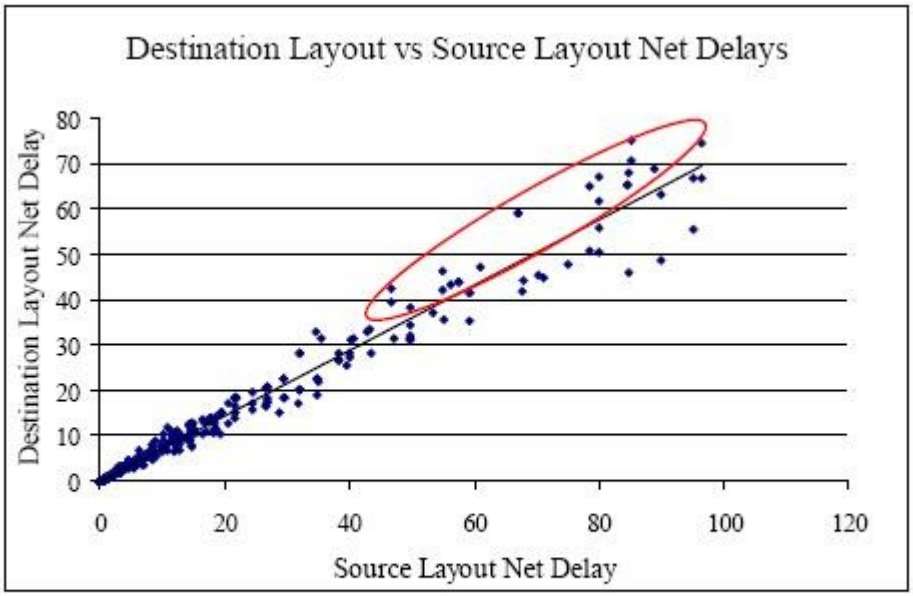


Figure 23: Destination Layout vs. Source Layout Net Delays. Encircled data points represent nets of significant delays which have been badly scaled

Systems and Processes for Asymmetrically Shirking a VLSI Layout, US Patent, Kever et. al., 2006

This patent gives an overview of the problem of circuit shrinking and proposes an algorithm for remedying this problem. The authors note that typically shrinking is achieved by first reducing the design by a factor and then repairing all the design rule faults. They say that this can take many man-months for very large circuits and may require a complete re-engineering of the circuit. They propose an automation algorithm to do the porting. This algorithm is similar to the one proposed above. To deal with problems related to wire shrinking, they propose to shrink wire asymmetrically (i.e. different scaling factors for the x and y directions). They also propose moving and reorienting cells which cannot be connected based on the design rules.

A Novel VLSI Layout Fabric for Deep Sub-Micron Applications, Design Automation Conference, 1999,Sunil P. Khatri Amit Mehrotra Robert K. Brayton Alberto Sangiovanni-Vincentelli, Ralph H. J. M. Otten

This conference paper while not directly connected to migration was stumbled upon in the references of the first publication noted above and it provides some a good ideas for circuit design that may be valuable. It describes a gridding of the metal layers to form a mesh. The goal set forth by this paper is to lower the inter-wire capacitance and reduce the dependence of the delay of a given signal line on other signal lines. They noted that the delay can more than double (2.5:1) based on activity of nearby signal wires. It proposes doing this by laying out layers by a series of wires with alternating VSS, signal, and VDD all going in the same direction. Then next wire layer will be the same but run perpendicular. This not only would reduce the inter-wire signal dependent capacitance but also provide strong VSS and VDD lines throughout the circuit.

They designed a layouts using such a method in 0.1, 0.25, and 0.05 micron processes and then used a program known as Space3D to calculate the inter-wire capacitance and thus the delay. They were able to reduce the inter-wire signal dependent capacitance to 2% thus greatly increasing circuit

predictability. This methodology also greatly reduced the inductance of wires because a signal return path is next to ever wire. However, all of this was at the cost of increased area.

Two-Dimensional Layout Migration by Soft Constraint Satisfaction, Proceedings of the Sixth International Symposium on Quality Electronic Design (ISQED'05), Q. Tang et. al.

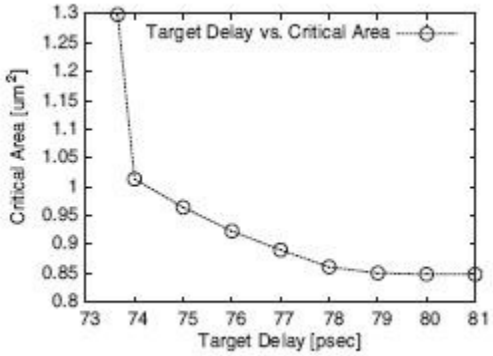
This paper discusses the problems with the commonly used (at that time) migration algorithms and proposes an improvement. The authors note that there are two main alogorithms. The first involves two successive 1D compactions and the second involves a single 2D compactions. The typical 1D compaction is fast but sometimes produces poor results. The 2D compactions tends to be much slower but tends to produce higher quality compaction. They proposed somewhat of a hybrid in an attempt to get the best qualities of both algorithms.

The process starts with very soft constraints on the parameters and then does "interative 1D compactions" each time tightening the constraints on the layout. That is to say at each iteration, size parameters are given a range and the size is optimized by repeated tightening of this range. They used this method to migrate several library components from a 1.2 to a 0.25 micron technology. They were able to achieve reduced size compared to the 1D compaction with a maximum runtime of 4 times that of the 1D compaction.

Timing-Driven Cell Layout De-Compaction for Yield Optimization by Critical Area Minimization, Proceedings of the conference on Design, automation and test in Europe: Proceedings, 2006 ,Tetsuya Iizuka, Makoto Ikeda, and Kunihiro Asada,

This paper proposes a method for decompaction with a goal to optimize the delay and yeild for a given cell area. They note that in order to increase yield it is often a requirement to increase the circuit area. However, doing so can sometimes produce very poor results for circuit performance. The method proposed by this paper, proposes to minimize the critical area of the circuit not the total area necessarily. The critical area is defined as the area in which a spot defect must be in order to produce a circuit fault. The method begins by first setting a delay constraint and they decompacting the circuit. The critical area is then successively minimized. In order to accurately assess the delay produced by their new circuits, they also proposed a new delay model. They took the traditional Elmore delay model and added an error factor which accounted for the change in capacitance with change in slew rate.

To demonstrate their findings, they optimized a 90nm library set and were able to show a fundemental relationship between critical area and delay. A graph of this can be seen below.



Calligrapher: A New Layout Migration Engine Based on Geometric Closeness,IEEE 2004,Fang Fang, Jianwen Zhu

This paper gives a brief overview of the problems associated with process migration and then proposes two new methods for migrating. The authors first give some motivation to migration rather than the purchase of new libraries. They note that it virtually eliminates the need for great library overhaul and it does eliminate the huge cost of new libraries. However, authors note that many process migration algorithms only take note of the design rules and thus end up ignoring many of the advanced concepts that went into the original library design. This paper proposes to resolve this using two new algorithm components. In one, they create a concept of geometric closeness whereby the space in a design is recognized as a layer and the program uses/values this to preserve likeness to the original design. The hope is that this likeness means that the new circuit preserves the designers concepts. In the other they propose a new constraint generation method that increases in run time linearly with complexity rather than the best previously proposed method with increases quadratically. They demonstrated the use of these concepts by the migration of several cells.

Current Status and Outlook

After discussing the proposed project with Professor Calhoun, we were left with a number of questions to answer in terms of how to best begin and what was the best path forward. The first goal was to familiarize ourselves with SKILL and PyCell and get a feel for its format, language, and capabilities. To this end, we first completed the SKILL tutorial from the class wiki and examined the script from this tutorial to understand how it worked. We then downloaded, installed and inspected PyCell. At this point, it was decided to begin by focusing our efforts on SKILL because the resources (tutorial, online, other students) for this language were much greater.

With a basic understanding of our options, we formed an initial plan. The plan would be to write a C++ program that would take in a layout (in Cadence or PyCell format) and then either produce a new layout. The goal was to write the algorithm only once in language which was well understood rather than in two languages which must be learned. To test and demonstrate this, a C++ program was written that imported all of the polygons from a Cadence layout text file export into a storage class. The class was then outputted to a new file in the same format. This demonstrated the possibility of this approach. With this complete, we reassessed the problem. We determined that this approach had some major limitations in that it lacked the some of the major pre-written algorythms we would need to fully implement this as a solution, namely the DRC and LVS checks. While implementing these in C++ is not a insurmountable problem, this hurdle seemed significantly greater than the double implementation of the algorithm.

We then decided to begin with a simple SKILL implantation of the program. We met with Stuart Wooters and Randy Mann to discuss their opinions on the most efficient path forward. After these meetings, a basic near future plan forward was decided upon:

1. Form a tentative list of goals for completion of the project.
2. Form a plan for the final algorithm to give a roadmap for script completion.
3. Begin learning SKILL by first producing a layout made entirely by SKILL script (this is also a proof of concept step).
4. Complete the tutorial on parametrized Cadence layouts (Stuart let us know it was there).
5. Based on this experience, either hand draw a parametrized inverter and NAND in standard cell format using Cadence in a format that will be easily ported (i.e. straight runs with as few angles as possible) or plan to produce SKILL include files for each library component in place of a physical layout. Henceforth, both of these implementations will be known as layouts.
6. Successively implement the porting algorithm for this layout starting with resizing, then resizing and LVS check, then resizing and LVS check, ect.
7. After each successive implementation, improve both the script and/or our layouts.
8. Use what we have learned to produce a similar PyCell Library and script.

Towards this end, we first formed our goals which we state as:

We will produce 3-5 representative library objects and a script for both PyCell and SKILL which include the porting and optimization algorithms and use these scripts to prove proof of concept by porting to 2-4 technologies. Optimization will include at least size optimization.

We next produced a plan for the final algorithm which can be seen below as a flow diagram in figure 1. Then we created a our proof of concept SKILL script which can be seen as attachment 1. Finally, we completed the parametrized layout tutorial. Next on the agenda is the decision to move forward in a SKILL script only format or a parametrized cell format.

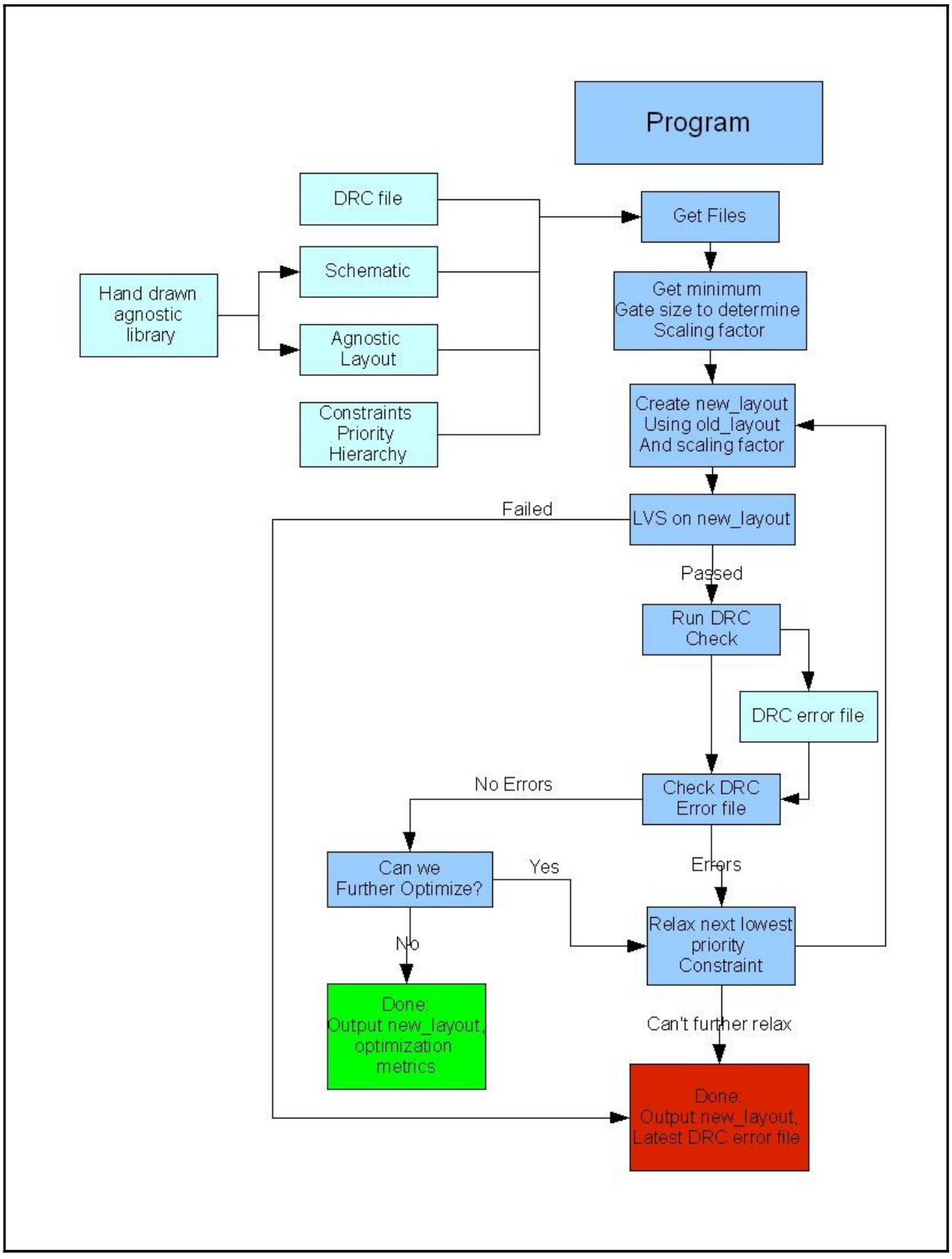

Program
DRC file
Schematic
Hand drawn agnostic library
Agnostic Layout
Constraints Priority Hierarchy
Get Files
Get minimum Gate size to determine Scaling factor
Create new_layout Using old_layout And scaling factor
LVS on new_layout
Failed
Passed
Run DRC Check
DRC error file
Check DRC Error file
No Errors
Errors
Can we Further Optimize?
Yes
No
Relax next lowest priority Constraint
Can't further relax
Done: Output new_layout, optimization metrics
Done: Output new_layout, Latest DRC error file

Figure 1: Plan for the final algorithm. Files are in light blue (light gray), methods are in darker blue (darker gray), outputs are in green and red (darkest grays).